

COMMUNICATION SERVER FOR AN INSTRUMENT MANAGEMENT SYSTEM

Field of the Invention

[0001] The present invention relates generally to a documentation system for tracking medical instruments, and more particularly to a communication server for interfacing an antimicrobial treatment device with an instrument management system.

Background of the Invention

[0002] Medical instruments (e.g., surgical instrument, bowls, basins, endoscopes, light cables, heart pacing cables, video cameras, microsurgical and eye instruments, ophthalmic lenses, internal and external defibrillator paddles and dilators) used by healthcare organization are typically subject to an instrument management cycle comprised of the following repeated steps: (1) instrument use, (2) instrument cleaning, (3) instrument storage, and (4) instrument sterilization. Moreover, the medical instruments may be also undergo periodic maintenance, repair, and replacement.

[0003] Healthcare organizations require quick, efficient and safe instrument reprocessing procedures. In order to improve patient safety and outcomes, there is a need to validate these reprocessing procedures. Accordingly, computer-based instrument tracking systems (or materials management systems) have been developed to facilitate the planning, control and documentation of antimicrobial treatment of medical instruments. An example of a typical instrument tracking system is the Sterile Processing Microsystem (SPM) by Materials Management Microsystems.

[0004] One drawback to existing instrument tracking systems is the inability to communicate directly with antimicrobial treatment devices (e.g., sterilizers and washers) from a wide variety of manufacturers. In some cases, hospital personnel must manually enter data into the instrument tracking system from the antimicrobial treatment devices.

[0005] The present invention addresses these and other drawbacks of prior art systems to provide a communication server for interfacing an antimicrobial treatment device with an instrument management system.

Summary of the Invention

[0006] In accordance with the present invention, there is provided an information management system for tracking instruments, comprising: (1) at least one instrument tracking client, each instrument tracking client installed on a respective general purpose computer; (2) at least one antimicrobial treatment device; and (3) a communication server interface in communication with the at least one instrument tracking client and the at least one antimicrobial treatment device, said communication server interface programmed to: (a) request data from the at least one antimicrobial treatment device, (b) transmit requests from at least one instrument tracking client, (c) receive data from the at least one antimicrobial treatment device, (d) receive requests from the at least one instrument tracking client, and (e) transmit data to the at least one instrument tracking client.

[0007] In accordance with another aspect of the present invention, there is provided a communication server interface in communication with at least one instrument tracking client and at least one antimicrobial treatment device, said communication server comprising: (a) means for requesting data from the at least one antimicrobial treatment device, (b) means for transmitting requests from at least one instrument tracking client, (c) means for receiving data from the at least one antimicrobial treatment device, (d) means for receiving requests from the at least one instrument tracking client, and (e) means for transmitting data to the at least one instrument tracking client.

[0008] An advantage of the present invention is the provision of a communication server that permits antimicrobial treatment devices from different vendors to communicate with an instrument tracking system.

[0009] Another advantage of the present invention is the provision of a communication server that facilitates communication with instrument tracking software from a wide variety of vendors.

[0010] Still another advantage of the present invention is the provision of a communication server that communicates using an industry standard communications protocol.

[0011] A still further another advantage of the present invention is the provision of a communication server that facilitates validation of instrument reprocessing procedures.

[0012] Still another advantage of the present invention is the provision of a communication server that facilitates quick identification of malfunctioning, or improperly configured, antimicrobial treatment devices.

[0013] Yet another advantage of the present invention is the provision of a communication server that facilitates paperless record keeping for sterilization and decontamination documentation.

[0014] These and other advantages will become apparent from the following description of a preferred embodiment taken together with the accompanying drawings and the appended claims.

Brief Description of the Drawings

[0015] The invention may take physical form in certain parts and arrangement of parts, a preferred embodiment of which will be described in detail in the specification and illustrated in the accompanying drawings which form a part hereof, and wherein:

[0016] FIG. 1 is a block diagram of an instrument management system including a communication server, according to a preferred embodiment of the present invention;

[0017] FIG. 2 is a first flow diagram illustrating the flow of function calls between instrument tracking clients and the communication server; and

[0018] FIG. 3 is a second flow diagram illustrating the flow of function calls between instrument tracking clients and the communication server.

Detailed Description of a Preferred Embodiment

[0019] Referring now to the drawings wherein the showings are for the purpose of illustrating a preferred embodiment of the invention only and not for purposes of limiting same, FIG. 1 shows a block diagram of an instrument management system 10. Instrument management system 10 includes a plurality of workstations 20 with associated instrument tracking clients 24, an instrument tracking server 30, a communication server interface 40, and antimicrobial treatment devices 80. It should be appreciated that "antimicrobial treatment devices," as used herein, includes, but is not limited to, devices for carrying out washing, sterilization, disinfection, decontamination, inactivation, and sanitization processes. The

components of instrument management system 10 communicate via a communications network 12. Communications network 12 may include a private communications network, and/or a public communications network, such as the Internet.

[0020] Workstations 20 may take the form of a general purpose computer, such as a conventional personal computer (PC) system, including an input unit (e.g., keyboard, keypad, barcode scanner or touchscreen) and an output unit (e.g., a video or LCD display unit or printer). Workstations 20 may be physically located throughout a healthcare facility, as well as locations remote from the healthcare facility. For instance, workstations 20 may be located at: (1) a central services or central supply (CS) department (also known as central processing department (CPD)), where instruments are decontaminated, and prepared and packaged; (2) an operating room (OR); (3) a maintenance and repair facility; (4) administrative offices; and (5) accounting offices.

[0021] Instrument tracking server 30 is a network server computer that provides services to instrument tracking clients 24 on workstations 20. Instrument tracking server 30 and instrument tracking clients 24 comprise an instrument tracking system (ITS).

[0022] Antimicrobial treatment devices 80, include, but are not limited to a first sterilizer 82, a second sterilizer 84 and a washer disinfectant 86. Antimicrobial treatment devices 80 may reduce microbial contamination by means of treatment agents (e.g., steam, ethylene oxide gas, peracetic acid and vaporous hydrogen peroxide), hot water, rinsing and drying.

[0023] Communication server interface 40 is comprised of a general purpose computer 45, such as conventional personal computer (PC), and a communication server 50. Communication server 50 is a server software program installed on computer 45 that serves instrument tracking clients 24. In this regard, communication server 50 sends requests for data, and fulfills requests for data, as will be described in detail below. Communication server 50 also operates to store data at computer 45. It should be understood that communication server 50 may alternatively be located on one of workstations 20, rather than computer 45.

[0024] In accordance with a preferred embodiment of the present invention, communication server 50 is implemented as a Component Object Model (COM), a COM+, and a Distributed Component Object Model (DCOM).

[0025] Component Object Model (COM) is a Microsoft® specification that defines the interaction between components in a Windows environment. A component is a self-contained coded module that provides some service to other components in an object-oriented environment. COM provides a set of interfaces allowing clients and servers to communicate within the same computer.

[0026] COM+ is both an object-oriented programming architecture and a set of operating system services. COM+ adds to COM additional system services for application components while they are running, such as notifying them of significant events or ensuring they are authorized to run. Among the services provided by COM+ are: (a) an event registry that allows components to publish the possibility of an event and other components to subscribe to be notified when the event takes place, (b) interception of designated system requests for the purpose of ensuring security, and (c) queues of asynchronously received requests for a service.

[0027] DCOM is the network version of COM that allows objects running in different computers attached to a network to interact. Accordingly, DCOM allows independent software components to communicate between different computers connected to the network. DCOM has several benefits, including, but not limited to: compatibility with multiple operating system platforms (e.g., Microsoft Windows, UNIX, Linux, Open VMS, and Solaris), compatibility with multiple software platforms (e.g., Visual Basic, C, C++, Java, Object Pascal, and COBOL), compatibility with multiple network protocols (e.g., TCP/IP, HTTP, and Novell), and allows implementation of security features to prevent unauthorized interfacing with the communication server software.

[0028] DCOM supports remoting objects by running on a protocol called the Object Remote Procedure Call (ORPC). The ORPC layer is built on top of distributed computing environment's Remote Procedure Call (RPC), and interacts with COM run-time services. A DCOM server is a body of code that is capable of serving up objects of a particular type at runtime. Each DCOM server object can support multiple interfaces each representing a different behavior of the object. A DCOM client calls into the exposed methods of a DCOM server by acquiring a pointer to one of the server object's interfaces. The client object then starts calling the server object's exposed methods through the acquired interface pointer, as if the server object resided in the client's address space.

[0029] Communication server 50 allows data from antimicrobial treatment devices 80 to be integrated into an instrument tracking client 24 (through the use of COM, COM+ and DCOM technology). In this regard, instrument tracking client 24 can directly monitor antimicrobial treatment devices 80 through communication server 50. For instance, instrument tracking client 24 can obtain cycle data including, but not limited to: (1) temperatures, (2) pressures, (3) cycle time, (4) operator name, (5) lot number associated with a specific load and operator name, (6) cycle start time, (7) total cycle time, and (8) cycle type.

[0030] Conventional instrument tracking systems may provide several functions, including, but not limited to: build and manage a master database of a healthcare facility's instrument inventory; track instrument sets; determine the correct type and quality of instruments and trays required allowing optimization of instrument set inventories; streamline or standardize instrument sets; locate needed instruments and trays; generate pick lists and service tickets; identify and track lost instruments and those needing repair or refurbishment; automation of sterilizer loads; manage staffing resources; improve instrument set turnaround time; create reports on inventory orders, repairs, needs projections, instrument set comparisons, productivity, sterilizer loads, and purchasing; facilitate staff training using online instrument and set images; track mobile equipment; and interface with OR scheduling software.

[0031] In accordance with a preferred embodiment of the present invention communication server 50 is compatible with a standard messaging protocol for clinical and administrative data. Preferably, the standard messaging protocol is HL7. The HL7 messaging standard promotes the exchange, management and integration of data that support clinical patient care and the management, delivery and evaluation of healthcare services.

[0032] Operation of instrument management system 10 will now be described in detail. Articles to be, or that have been, treated together using a given antimicrobial treatment process are referred to as a "Load." A "Pack" is any pre-prepared instrument set, or alternatively may be a single instrument. A "Pack List" is created and maintained by the instrument tracking client 24. A "Cycle" refers to a defined sequence of operational steps of an antimicrobial treatment process. "Cycle Time" refers to the total elapsed time of a Cycle.

[0033] A “Load Number” is typically an integer that identifies an accumulation of Packs that have been placed together in an antimicrobial treatment device for a particular “Lot Number.” The Load Number is the sequential number of the Load. The Load Number increases by one each time a Cycle is executed. Load Numbers increase in this manner until midnight of each day, after which they reset to 01. “Open Load” refers to any Load that has not been processed, while a “Closed Load” refers to a processed Load. Pack information for a Closed Load is not typically edited or changed, but may be viewed for reports.

[0034] Sterilizer loads are tracked by automatic assignment of Lot Numbers by the instrument tracking client 24. For instance, the Lot Number may take the form of a 10-digit designation consisting of the date, an antimicrobial treatment device number, and a Load Number. As an example, Lot Number 083003-0201 refers to a date of August 30, 2003, an antimicrobial treatment device number (02), and a Load Number (01). The Load Number increases by one each time a Cycle is selected. Load Numbers increase in this manner until midnight of each day, after which the Load Number resets to '01.' An “Open Lot Number” refers to any Lot Number(s) assigned to Loads that have not been processed.

[0035] In accordance with a preferred embodiment, communication server 50 provides at least one of five different Lot Control modes, defined as follows:

[0036] (1) CPD with Operator: requires that the Load contents data be entered at the client workstation 20, and that the operator enter an operator identification (ID) code at the antimicrobial treatment device at the start and end of the cycle. A Lot Number is printed on a cycle tape printout.

[0037] (2) CPD without Operator: requires that the Load contents data be entered at a client workstation 20, but the operator code at the antimicrobial treatment device 80 is not required. A Lot Number is printed on a cycle tape printout.

[0038] (3) OR with Operator: does not require Load contents data be entered at a client workstation 20. The operator must enter an operator ID code at the sterilizer at the start and end of each cycle. A Lot Number is printed on a cycle tape printout.

[0039] (4) OR without Operator: does not require Load contents data to be entered at a client workstation 20, and does not require an operator ID code be entered at antimicrobial treatment device 80. A Lot Number is printed on a cycle tape printout.

[0040] (5) Non-Instrument Tracking Mode: does not require Load contents data entry or operator ID code entry. The Lot Number is not passed back and forth to antimicrobial treatment device 80, but the device 80 is still online as far as communication server 50 is concerned. Cycle tape information is gathered and may be displayed by client 24 on a status display screen.

[0041] It should be understood that load building and load management are functions managed by instrument tracking clients 24. By way of example, and not limitation, Loads for an antimicrobial treatment device 80 may be processed in the following manner in the “CPD with Operator” Lot Control Mode:

[0042] 1) An antimicrobial treatment device 80 is selected and associated with a Load.

[0043] 2) The operator logged into client 24 is associated with the Load.

[0044] 3) A new Lot Number is created and assigned.

[0045] 4) The Load is built by the operator by selecting from a list of item descriptions, set descriptions or packs.

[0046] 5) The Load is closed. The operator verifies the contents of the Load matches the desired Lot Number, and verifies the Load is associated with the selected antimicrobial treatment device 80.

[0047] 6) The Lot Number is sent to the selected antimicrobial treatment device 80.

[0048] 7) The Load is placed in the selected antimicrobial treatment device 80.

[0049] 8) The desired Lot Number is selected at the selected antimicrobial treatment device 80.

[0050] 9) The desired cycle is selected.

[0051] 10) The operator ID code is entered if in CPD with operator mode.

[0052] 11) Client 24 parses “print and display data” from the selected antimicrobial treatment device 80 to determine the status of the Load (i.e., running or complete). The status information is sent to all clients 24, including Lot Number, Operator and a number associated with the selected antimicrobial treatment device 80. The status of the Load may be displayed at workstation 20.

[0053] 12) Client 24 sends the Lot Number to the selected antimicrobial treatment device 80, where it is displayed with any others assigned to the same antimicrobial treatment device 80.

[0054] 13) After the Load is processed by the selected antimicrobial treatment device 80, post cycle data may be added to the Load, such as the results of biological or chemical indicators.

[0055] 14) The operator enters their operator code, and removes the Load from the antimicrobial treatment device 80.

[0056] 15) Client 24 deletes the Lot Number from the antimicrobial treatment device 80.

[0057] By way of example, and not limitation, a typical instrument tracking client 24, may display: (1) current date; (2) all Loads for the current date; (3) state of the Load; (4) an empty status (i.e., the Load is created but contains no items); (5) a loading status; (6) a running status (i.e., Load being processed by an antimicrobial treatment device); (7) a complete status (i.e., antimicrobial treatment device 80 has finished processing the Load); (8) a canceled status; and (9) an indication of whether the Load contains a dart or a biological indicator.

[0058] It should be appreciated that antimicrobial treatment devices 80 may support tracking of treated articles by allowing a Physician ID, Patient ID or Procedure ID to be tied to a particular Load. Accordingly, server 50 supports the downloading of the of a Physicians list, Procedure ID or Patient ID.

[0059] In accordance with a preferred embodiment, communication server 50 supports the various Lot Control modes (described above), as well as various operations of clients 24, through the functions of: SetMessage, GetMessage, OnSetMessage, and OnGetMessage. It should be appreciated that other communication protocols may also be implemented.

[0060] The SetMessage function allows any client 24 to issue a command to communication server 50. Communication server 50 then responds to the command, but does not return any data to client 24. In some cases, communication server 50 sends a command to an antimicrobial treatment device 80 in response to a SetMessage function.

[0061] The GetMessage function allows any client 24 to request data from communication server 50. Communication server 50 responds to the request by transmitting the requested data to client 24.

[0062] The OnSetMessage function allows communication server 50 to notify all clients 24 of updated information. Clients 24 determine whether the updated information pertains to it, and how to process the information.

[0063] The OnGetMessage function allows communication server 50 to request a client 24 to send information to communication server 50. Communication server 50 can then forward the information to an antimicrobial treatment device 80 that needs the information. Client 24 may send the requested information to communication server 50 using the SetMessage function.

[0064] In accordance with a preferred embodiment of the present invention, the SetMessage, GetMessage, OnSetMessage and OnGetMessage functions have an argument that is preferably an HL7 message. The HL7 message is a data string including a message name, field names, and field values.

[0065] As indicated above, the SetMessage function allows any client 24 to issue a command to communication server 50. Supported Messages and Fields for the SetMessage Function are listed and described in TABLE I.

TABLE I: SetMessage Function - Messages and Fields

| MessageName | Fields | Description |
|---------------------------|--|--|
| AddLotToDevice | LotNumber, LotNumberFlag | Client 24 sends this message either by its own choosing or after receiving an OnGetMessage. Server 50 extracts the DeviceNumber from the LotNumber and commands Device 80 to add the LotNumber. Server 50 may use the LotNumberFlag for graphical user interface (GUI) display purposes and to decide whether to issue a regular or special ADD_LOT command to the device. |
| AddOperatorToDevices | OperatorName, OperatorCode, OperatorOffset, BarCodeNumber | Client 24 sends this message by its own choosing. Server 50 commands all Devices 80 to add the operator. |
| AddPhysicianToDevices | PhysicianName, Offset, BarCodeNumber | Client 24 sends this message by its own choosing. Server 50 commands all Devices 80 to add the physician. |
| DeleteLotFromDevice | LotNumber, LotNumberFlag | Client 24 sends this message by its own choosing. Server 50 extracts the DeviceNumber from the LotNumber and commands the Device 80 to delete the LotNumber. |
| DeleteOperatorFromDevices | OperatorOffset | The Client 24 sends this message by its own choosing. Server 50 commands all Devices 80 to delete the operator. |
| LoadContentsChanged | LotNumber | The Client 24 sends this message by its own choosing (when it detects the change). Server 50 issues an OnSetMessage to all other connected Client 24 to notify them. If they care about this LotNumber, they can refresh their load information. |
| LoadStatusChanged | StatusIndicator | Client 24 sends this message by its own choosing (when it detects the change). Server 50 issues an OnSetMessage to all other connected Clients 24 to notify them. |

| TABLE I: SetMessage Function - Messages and Fields | | |
|--|---|---|
| MessageName | Fields | Description |
| SendControlCodeToDevice | DeviceNumber, ControlCode | Client 24 sends this message by its own choosing. Server 50 sends the ControlCode to the Device 80 as a command. |
| SendPackInfoToDevice | DeviceNumber, ItemDescription, PackCode | Client 24 sends this message either by its own choosing or after receiving an OnGetMessage. Server 50 commands the Device 80 to add the ItemDescription and PackCode. |
| SystemConfiguration | FacilityName, NumberOfSterilizers, Field4, DateFlag, NumberOfBioReadings | Client 24 sends this message by its own choosing. Server 50 updates its registry with the information, as if a user had entered it in the Configure System form, and issues an OnSetMessage to all other connected Clients 24 to notify them. |

[0066] As discussed above, the GetMessage function allows any client 24 to obtain data from communication server 50. Supported Messages and Fields for the GetMessage function are listed and described in TABLE II.

| TABLE II: GetMessage Function - Messages and Fields | | |
|---|------------------------------|--|
| MessageName | Fields | Description |
| GetDeviceData | DisplayType, DeviceNumber | After receiving a NewDeviceData notification, any interested Client 24 sends this message. Server 50 returns the requested data. |
| GetLotNumberFromDevice | DeviceNumber | A Client 24 sends this message by its own choosing. Server 50 parses the Device 80 display data to find the LotNumber and returns it to the Client 24. |

[0067] As indicated above, the OnSetMessage function allows communication server 50 to notify all clients 24 of updated information. Supported Messages and Fields for the OnSetMessage function are listed in TABLE III.

| TABLE III: OnSetMessage Function - Messages and Fields | |
|---|---|
| MessageName | Fields |
| LoadContentsChanged | LotNumber |
| LoadStatusChanged | StatusIndicator |
| NewDeviceData | DisplayType, DeviceNumber, DisplayText (only when Display Type is zero) |
| DeviceConfiguration | DeviceNumber, DeviceName, DeviceModel, DeviceType, LotControl |
| SystemConfiguration | FacilityName, NumberOfSterilizers, Field4, DateFlag, NumberOfBioReadings |

[0068] As indicated above, OnGetMessage allows communication server 50 to request a Client 24 to send a message back to communication server 50. Communication server 50 forwards the information to Device 80. Supported Messages and Fields for the OnGetMessage function are listed and described in TABLE IV:

| TABLE IV: OnGetMessage Function - Messages and Fields | |
|--|-----------------------------|
| MessageName | Fields |
| AddLotToDevice | DeviceNumber, LotNumberFlag |
| SendPackInfoToDevice | DeviceNumber, BarCodeNumber |

[0069] Set forth below is a summary of field names, including data type and a brief description.

| Field Name | Data Type | Range and Definition |
|---------------|-----------|--|
| BarCodeNumber | | Text, usually digits, that is a value of a scanned barcode |
| ControlCode | Integer | 0 – 255 |
| DateFlag | Integer | One of the following integers: <ul style="list-style-type: none"> • 1 for USA: mm/dd/yy • 2 for ANSI: yy.mm.dd • 3 for Britain/France: dd/mm/yy • 4 for Germany: dd.mm.yy • 11 for Japan: yy/mm/dd • 12 for ISO: yymmdd |
| DeviceModel | Integer | For example, one of the following integers for the corresponding devices 80: <ul style="list-style-type: none"> • 1 for <i>Stage 3</i> • 2 for <i>Stage 2</i> • 3 for <i>Non 3000</i> • 4 for <i>Century</i> • 5 for <i>System 2S</i> • 6 for <i>System 1</i> • 7 for <i>Hoplab 130</i> |
| DeviceName | Char | The name of the Device 80 (e.g., VAC 01) |
| DeviceNumber | Integer | 1 – 30 |
| DeviceType | Integer | For example, one of the following integers: <ul style="list-style-type: none"> • 1 for <i>Steam</i> • 2 for <i>Gas</i> • 3 for <i>444</i> • 4 for <i>777</i> • 5 for <i>3400</i> • 6 for <i>PIWS</i> • 7 for <i>3000 SL</i> • 8 for <i>Floorloader</i> • 9 for <i>3017</i> • 10 for <i>Plasma</i> • 11 for <i>PACid</i> • 12 for <i>Hoplab 130</i> |

| Field Name | Data Type | Range and Definition |
|-----------------|-----------|--|
| DisplayText | Char | Text appearing in the display screen associated with Device 80 |
| DisplayType | Integer | <ul style="list-style-type: none"> 0 for Display data 1 for Printout data |
| FacilityName | String | Name of healthcare facility |
| Field4 | Integer | <ul style="list-style-type: none"> 0 for Cost 1 for Time (Labor Hours) |
| ItemDescription | String | Describes a Pack, (e.g., a heart pan set) |
| LotControl | Integer | <ul style="list-style-type: none"> 72 for Offline 73 for OR without OPER 74 for OR with OPER 75 for CPD without OPER 76 for CPD with OPER |
| LotNumber | Char | mmddyyXDDNN, where: mm = two-digit month (01 = Jan, ..., 12 = Dec) dd = two-digit day (01, ..., 28 or 29 or 30 or 31) yy = two-digit year (03 = 2003, ..., 99 = 2099) X = normally a dash (-), but if it is in a cycle, it will be 1 (for LEAK), 2 (for DART WARM), or 3 (for DART) DD = two-digit DeviceNumber (01, ..., 30) NN = two-digit Load Number (01, ..., 99) |
| LotNumberFlag | Integer | One of the following integers: <ul style="list-style-type: none"> 0 for regular lot (to add to the list) 1 for special lot (to issue real-time when the device requests it) |

| Field Name | Data Type | Range and Definition |
|---------------------|-----------|--|
| MessageName | | <p>For Server - SetMessage():</p> <ul style="list-style-type: none"> • AddLotToDevice • AddOperatorToDevices • AddPhysicianToDevices • DeleteLotFromDevice • DeleteOperatorFromDevices • LoadContentsChanged • LoadStatusChanged • SendControlCodeToDevice • SendPackInfoToDevice • SystemConfiguration <p>For Server - GetMessage():</p> <ul style="list-style-type: none"> • GetDeviceData • GetLotNumberFromDevice <p>For Client - OnSetMessage():</p> <ul style="list-style-type: none"> • LoadContentsChanged • LoadStatusChanged • NewDeviceData • DeviceConfiguration • SystemConfiguration <p>For Client - OnGetMessage():</p> <ul style="list-style-type: none"> • AddLotToDevice • SendPackInfoToDevice |
| NumberOfBioReadings | Integer | 1 or 2 |
| NumberOfSterilizers | Integer | 1 – 30 |
| OperatorCode | Char | Text |
| OperatorName | Char | Text |
| OperatorOffset | Integer | 0 – 9999 |
| PackCode | Char | Text |
| PhysicianName | Char | Text |
| PhysicianOffset | Integer | 0 – 9999 |

| Field Name | Data Type | Range and Definition |
|-----------------|-----------|--|
| StatusIndicator | | <p>One of the following:</p> <ul style="list-style-type: none"> • <i>CLL</i><LotNumber> to tell Clients 24 to close any open forms that are open for the LotNumber • <i>CST</i><LotNumber><OperatorName><DeviceNumber> to tell the Clients 24 a cycle started (each field is fixed length, space-padded to the right) • <i>LDS</i> to tell the Clients 24 to refresh their device load control displays |

[0070] Referring now to FIGS. 1 and 2, there is shown a flow diagram illustrating examples for SetMessage, GetMessage, OnSetMessage and OnGetMessage functions. It should be understood that Client1 and Client2 refer to two different clients 24.

[0071] It is contemplated that antimicrobial treatment device specific messages may be developed and adopted as HL7 standard messages. These messages may include, but are not limited to:

[0072] (a) Sterilizer Command Message (SCM), a message sent by instrument tracking client 24 to program a sterilizer;

[0073] (b) Sterilizer Data Message (STD), a message sent from instrument tracking client 24 to obtain data from the sterilizer;

[0074] (c) Sterilizer Acknowledge Message (SAK), a message for the sterilizer to acknowledge receipt of a message;

[0075] (d) Sterilizer Notification Message (SNT), a message sent from the sterilizer to notify all instrument tracking clients 24 of updated information; and

[0076] (e) Sterilizer Request Message (SRQ), a message sent from the sterilizer to an instrument tracking client 24 to obtain data from the instrument tracking client 24.

[0077] Other modifications and alterations will occur to others upon their reading and understanding of the specification. It is intended that all such

modifications and alterations be included insofar as they come within the scope of the invention as claimed or the equivalents thereof.